International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072

COVID-19 DETECTION USING X-RAY IMAGES BY USING CONVOLUTIONAL NEURAL NETWORK

M.Vidhya[1], Gunupati Gishma[2], Sarnu Venkata Sravya[3]
Assistant Professor[1], IV Year students[2,3]
Department of CSE
Panimalar Institute of Technology,Chennai.

---***---

Abstract - *- The coronavirus 2019(COVID-19), which first occur in wuhan city of China in December 2019, spread quickly around the world and became a plague. It is necessary to detect the positive cases as early as possible so as to prevent the further spread of this disease. Application of convolutional neural networks (CNN) techniques coupled with medical imaging can be helpful for the accurate detection of this disease. In this project a new model for automated COVID-19 detection using raw chest X-ray images is used. The model is developed to supply accurate diagnosis for binary classification (COVID vs No-Findings). Our model produced an accuracy of 98.44%. The sequential model was used to train a model using Keras and Tensorflow. We introduced convolutional layers and implemented different filtering on each layer.*

Key Words: *Convolutional neural networks, keras and Tensor flow, Flask* **,** *numpy, Image processing*.

1.INTRODUCTION

COVID-19 is a transmissible disease that was caused by the Severe Acute Respiratory Syndrome Coronavirus 2(SARS-CoV-2). The disease was first discovered in china and has spread among the world. As we know, on December 12, 2020 more than 12,847384 confirmed cases of COVID-19 and 55,587 confirmed deaths due to disease. Signs of infection include breathing problem, cough, fever. In more serious cases, the infection can cause severe acute respiratory syndrome, septic shock, multi organ failure and death.

Due to the regular increase in cases, the number of COVID-19 test kits available in the hospital is minimal.
Since an automated detection system is needed as an alternative diagnosis to prevent COVID-19 from spreading among people.Keras and Tensorflow were used to train a model using the sequential model. We created layers of convolutional neural networks and applied different filtering to each layer.

The objective of this project is improve COVID-19 detection accuracy from X-ray images of the chest. CNNs (Convolutional Neural Networks) are a form of deep neural network. The Convolutional neural network structure has two layers, one of which is the feature extraction layer, which connects each layer's input to the previous layer's local receptive fields and extracts the local feature. The positional relationship between the local feature and other features will be calculated once it has been extracted.
In this regard, we consider a system based on CNN, since CNN is a powerful feature extraction and classification methodology with excellent image classification recognition efficiency. Of course, in the case of medical image analysis, substantial diagnostic accuracy can be a primary goal alongside important findings, and in recent years, the discovery of critical facts in medical imaging has been driven by a CNN-based system, which encourages us to try.

2. EXISTING METHEDOLOGIES

2.1 SWAB TEST:-

The person administering the test will twirl a long stick with a soft brush on the end, similar to a pipe cleaner, up your nose for a few seconds. There will be a collection of secretions collected by the soft bristles for study.To get a good specimen, the swab must go back a long way, so cells and fluids must be extracted from the whole passageway that links the base of the nose to the back of the neck.

Fig.1: Swab Test

However, since the body isn't used to having an object in that place, it produces a slew of strange sensations. For one thing, it activates the lachrymal reflex, which means it will bring tears to your eyes if it's done correctly. I wouldn't go so far as to say it hurt, but it is uncomfortable. Since the swab will also touch the back of the throat, it may also trigger a gag reflex.

2.2 NASAL ASPIRATE:-

In nasal aspirate, secretions from the back of your nose and upper throat are collected using a swab. Sometimes, a suction device may be used to gently remove the secretions. This is known as nasal (or nasopharyngeal) aspirate.

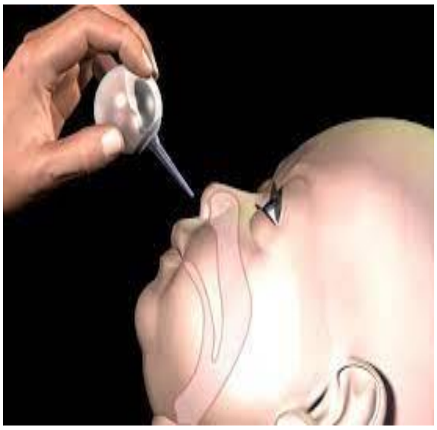

Fig.2: Nasal Aspirate

The secretions are send to a laboratory where they are grown. This makes it easier to identify which virus, bacteria or fungi are present. The results are sent back to your doctor who will use them to help diagnose what germs could be causing your symptoms.

2.3 SPUTUM TEST:-

A sputum test is also known as a sputum culture, is a test that your doctor may order when you have a respiratory tract infection or other lung-related disorder to determine what is growing in lungs.

Fig.3: Sputum Test

Sputum is a thick substance that accumulates when bacteria or fungi grows and multiplies in the lungs or bronchi. As it accumulates, the growing substance can make breathing more difficult and cause coughing.

3. PROPOSED SYSTEM

We have proposed an automatic prediction of covid-19 using a deep convolutional neural networks and Chest X-ray images. For this implementation, python programming language used with Tensor Flow Keras and OpenCV modules. We have deployed the previously trained CNN model in a web application using a python backend with a Flask web development framework. HTML and JavaScript are used as the frontend of the website. The proposed models will have end-to-end structure without manual feature extraction and selection methods. Chest X-ray is the best tool for the detection of covid-19.

Convolutional Neural Network (CNN) might seem intimidating for a beginner. However, this project will provide an overview of how to build a model to detect COVID-19 using Tensorflow and Keras.

The content of the project is listed as follows:

- Creating Dataset
- Data Preprocessing
- Training the CNN
- Webpage Creation

3.1 CREATING DATASET:

This experiment leveraging the data from Covid Chest X-Ray Dataset and Pneumonia dataset by Praveen from GitHub and Kaggle respectively. Using these two datasets, we have created a customized dataset as COVID19 Negative and COVID19 Positive.

3.2 DATA PREPROCESSING:

In this process, we are using two categories as 0 for COVID-19 Negative and 1 for COVID-19 Positive shown below:

{'Covid19 Negative': 0, 'Covid19 Positive': 1}

['Covid19 Negative', 'Covid19 Positive']

[0, 1]

Let us set the original image set to be S_1, which is composed of n images as

$$S_1=\{s_1(1), s_1(2),..., s_1(i),...\}$$

First, we compress the three channel color image to gray image, and get the grayscale image set S_2 as

$$S_2=G(S_1|BGR\rightarrow Grayscale)$$
$$=\{s_2(1), s_2(2),..., s_2(i),...\}$$

Resizing the gray scale into [100 100], since we need a fixed common size for all the images in the dataset and then appending the image and the label(categorized) into the list (dataset). Now, it will be normalized and saved as data and target.

3.3 TRAINING THE CNN:

Before we thought of training the CNN, we will load the data and target.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
model_1 (Model)              (None, 100, 100, 384)     11008
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 98, 98, 64)        221248
_________________________________________________________________
activation_1 (Activation)    (None, 98, 98, 64)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 49, 49, 64)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 47, 47, 32)        18464
_________________________________________________________________
activation_2 (Activation)    (None, 47, 47, 32)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 23, 23, 32)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 16928)             0
_________________________________________________________________
dropout_1 (Dropout)          (None, 16928)             0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               2166912
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 64)                8256
_________________________________________________________________
dropout_3 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 130
=================================================================
Total params: 2,426,018
Trainable params: 2,426,018
Non-trainable params: 0
```

Fig.4:Sequential Model Layers with parametres

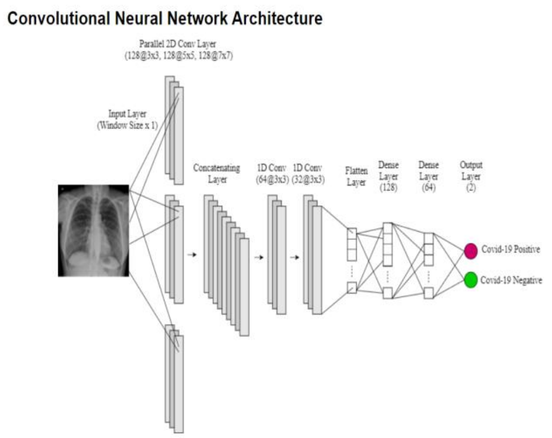

Fig.5: convolutional neural network architecture

Before training the model, we will set up data generators to read images from source folders instead of labeling the image one by one. Basically, ImageDataGenerator will label the images based on the directory the image is contained. It points to the sub-directory of the data. Grayscale normalization should be done in order for CNN to converge faster by using rescale parameter. The normalization transforms the image pixel from [0…255] to [0…1].

Create two generators namely train_data and train_target. Where train_data points to the sub-directory of train data and train_target points to the sub-directory of evaluation data.

Then, train the model for 15 epochs with 15 steps on every epoch.

```
Train on 1555 samples, validate on 173 samples
Epoch 1/15
1555/1555 [==============================] - 37s 24ms/step - loss: 0.4550 - accuracy: 0.7576 - val_loss: 0.1145 - val_accuracy:
0.9595
Epoch 2/15
1555/1555 [==============================] - 26s 17ms/step - loss: 0.1954 - accuracy: 0.9357 - val_loss: 0.0656 - val_accuracy:
0.9769
Epoch 3/15
1555/1555 [==============================] - 26s 17ms/step - loss: 0.1050 - accuracy: 0.9601 - val_loss: 0.0401 - val_accuracy:
0.9884
Epoch 4/15
1555/1555 [==============================] - 26s 17ms/step - loss: 0.0850 - accuracy: 0.9711 - val_loss: 0.0671 - val_accuracy:
0.9827
Epoch 5/15
1555/1555 [==============================] - 26s 17ms/step - loss: 0.0795 - accuracy: 0.9743 - val_loss: 0.0912 - val_accuracy:
0.9711
Epoch 6/15
1555/1555 [==============================] - 26s 17ms/step - loss: 0.0737 - accuracy: 0.9730 - val_loss: 0.0381 - val_accuracy:
0.9884
Epoch 7/15
1555/1555 [==============================] - 26s 17ms/step - loss: 0.0472 - accuracy: 0.9859 - val_loss: 0.0294 - val_accuracy:
0.9942
Epoch 8/15
1555/1555 [==============================] - 26s 17ms/step - loss: 0.0466 - accuracy: 0.9852 - val_loss: 0.0728 - val_accuracy:
0.9653
Epoch 9/15
1555/1555 [==============================] - 26s 17ms/step - loss: 0.0631 - accuracy: 0.9794 - val_loss: 0.0414 - val_accuracy:
0.9769
Epoch 10/15
1555/1555 [==============================] - 26s 17ms/step - loss: 0.0506 - accuracy: 0.9820 - val_loss: 0.0471 - val_accuracy:
0.9769
Epoch 11/15
1555/1555 [==============================] - 26s 17ms/step - loss: 0.0304 - accuracy: 0.9865 - val_loss: 0.0339 - val_accuracy:
0.9942

Epoch 12/15
1555/1555 [==============================] - 26s 17ms/step - loss: 0.0362 - accuracy: 0.9878 - val_loss: 0.0224 - val_accuracy:
0.9942
Epoch 13/15
1555/1555 [==============================] - 26s 17ms/step - loss: 0.0169 - accuracy: 0.9961 - val_loss: 0.0372 - val_accuracy:
0.9884
Epoch 14/15
1555/1555 [==============================] - 26s 17ms/step - loss: 0.0259 - accuracy: 0.9916 - val_loss: 0.0309 - val_accuracy:
0.9942
Epoch 15/15
1555/1555 [==============================] - 26s 17ms/step - loss: 0.0171 - accuracy: 0.9942 - val_loss: 0.0315 - val_accuracy:
0.9942
```

Fig.6: Training and Validation results

3.4 WEBPAGE CREATION:-

The deployment of the previously trained CNN model have done in a web application using a python back end with a flask web development frame work. The front end of the website is created with HTML and Javascript.

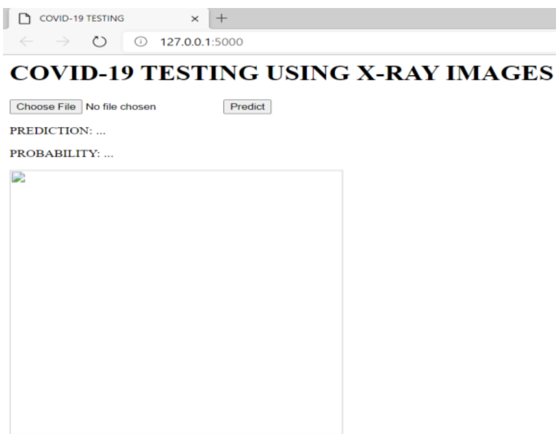

Fig.7: webpage

4. OUTPUT

In this process the front end and back end codes of this project allows the users to input an image and to determine whether the X-ray image is diagnosed with covid-19. It will give the prediction and probability when the user clicks the predict button after importing the input as X-ray image.
The following are the two prediction of this project:

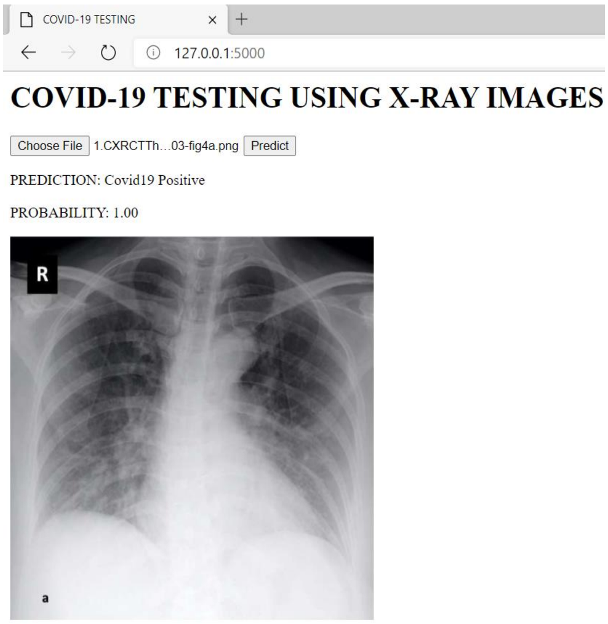

Fig.8: Prediction of COVID-19 Positive in Webpage

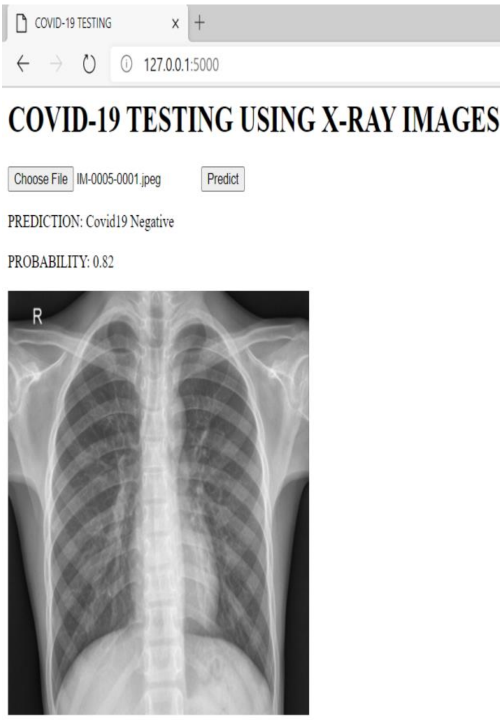

Fig.9: Prediction of COVID-19 Negative in Webpage.

5. CONCLUSION

In this project, we proposed a deep learning CNN model called sequential model for detecting corona virus disease(covid-19) from chest X-ray images. As we have seen, this model can effectively capture covid-19 features in the parallel layers of convolutional network, so it has an excellent performance compared to some well-known CNN architecture. Our experimental evaluation clearly shows that this model gives an accuracy of 98.84%. Using our proposed model, user can get the results of COVID-19 through webpage instantly. Therefore, this model can help to stop the spread of the pandemic with less cost and time.
As future work, we can use transfer learning or pre-trained CNN models to make our proposed model more robust. Also, we can develop mobile applications for easier usage which makes COVID-19 prediction more effective.

REFERENCES

[1] Liu, S., Satapathy, S.C., Zhang, YD. et al (2020). A five-layer deep convolutional neural network with stochastic pooling for chest CT-based COVID-19 diagnosis. Machine Vision and Applications 32, Article number: 14

[2] Ashad, Chowdhury, Md. Muhtadir & Kabir, Nihad&Rahman (2020). PDCOVIDNet: A Parallel-Dilated Convolutional Neural Network Architecture for Detecting COVID-19 from Chest X-Ray Images. 10.31224/osf.io/myp6c.

[3] Choi,W., Jeon, E., Kim, J., Lee, K. (2020). Evaluation of Scalability and Degree of Fine-Tuning of Deep Convolutional Neural Networks for COVID-19 Screening on Chest X-ray Images Using Explainable Deep-Learning Algorithm. J. Pers. Med. 10, 213.

[4] Boran Sekeroglu1 and Ilker Ozsahi2 (2020).Detection of COVID-19 from Chest X-Ray Images Using Convolutional Neural Networks SLAS TECHNOLOGY: Translating Life Sciences Innovation Volume 25, Issue 6.

[5] Arman Haghanifar, Mahdiyar Molahasani Majdabadi, Younhee Choi (2020). COVID-CXNet: Detecting COVID-19 in Frontal Chest X-ray Images using Deep Learning on 16 Jun 2020.

[6] Cesar, P., Soares, Lucas, P., Soares (2020). Automatic Detection of COVID-19 Cases on X-ray images Using Convolutional Neural Networks on on 2 Jul 2020.

[7] Yi Zhong (2020). Using Deep Convolutional Neural Networks to Diagnose COVID-19 From Chest X-Ray

Images on 19 Jul 2020.

[8] Chaimae Ouchicha, Mohammed Meknassi, Ouafae Ammor (2020). CVDNet: A novel deep learning architecture for detection of coronavirus (Covid-19) from chest x-ray images, Chaos, Solitons & Fractals,Volume 140, 110245, ISSN 0960-0779.

[9] Abdelgawad, A., Farhan Haque, F., Foysal Haque, K., Gandy, L. (2020). "Automatic Detection of COVID-19 from Chest X-ray Images with Convolutional Neural Networks," 2020 International Conference on Computing, Electronics & Communications Engineering (iCCECE), Southend, United Kingdom, 2020, pp. 125-130, doi: 10.1109/iCCECE49321.2020.9231235.

[10] Abbas Sharifi, Mohsen Ahmadi, Shayan Hassantabar (2020). 10. Diagnosis and detection of infected tissue of COVID-19 patients based on lung X-ray image using convolutional neural network approaches, Chaos, Solitons & Fractals, Volume 140, 110170, ISSN 0960-0779. https://doi.org/10.1016/j.chaos.2020.110170.